

PRICE \$2.00

## THE 80 NOTEBOOK

JULY 1980

ISSUE #4

.....

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation. THE 80 NOTEBOOK is not affiliated with Radio Shack or Tandy Corporation in any way.

.....

### NLOS/1 AND NLOS/2

#### A Natural Language Operating System for the TRS-80

What is natural language? For our purposes, natural language is common, ordinary English - the expression of facts using simple sentences. NLOS/1 is a system which allows the computer to "understand" the information conveyed to it through simple sentences and to answer questions concerning the information conveyed. This ability makes the system an excellent tool for the creation, management and inquiry to a conversational data base of facts and figures. It can also be an educational tool - as a study in artificial intelligence through an examination of the internal workings of the program, or by its reaction to math reading problems of various complexity. It can be an excellent tool in teaching English grammar, sentence structure and logical deductive reasoning to students, young and old alike. In any case, holding a conversation with your computer can be a lot of fun.

Let's see how the system accomplishes this and what its limitations are:

First, we have to get down to the basics of English. The system recognizes phrases grouped together in a sentence. A phrase is a group of one or more words that, together, convey a concept or identify an entity. These phrases can convey a subject, a verb, a preposition, a conjunction, a modifier or a question invoker. These represent the grammatical types that NLOS/1 can handle. You may ask - why not recognize individual words rather than phrases so that the conventional grammatical types - nouns, adjectives, verbs, adverbs, prepositions, conjunctions, interjections, and pronouns may be used? Well, recognizing nothing smaller than a phrase alleviates the problem of context usage. For example - "the President of the United States" contains adjectives, nouns and a preposition, yet the phrase identifies a single subject. Each grammatical type further conveys a type of information. Subjects identify a person (a who - "Tom", etc.), a place (a where - "New York City", etc.), or a thing (a what - "a banana", etc.).

Verbs convey the type of object clause they affect - in "Tom said nothing" and "Tom went home", etc., "said" acts on "what" and "went" acts on "where".

Every sentence must have a verb. Prepositions introduce subjects which show what ("as a clown"), when ("on my birthday"), where ("in town"), why ("to go home") and how ("by going home").

"Why" or "how" prepositional phrases are usually a combination of prepositions and verbs (such as why - "to go to"), while most other prepositional phrases consist of a single prepositional word.

Conjunctions show no type of information but are used to combine two adjacent subject phrases into a single subject clause. Because of this, conjunctions are restricted to "and" type conjunctive phrases. Also, only one conjunction may be used in each sentence analyzed by the system. The reason for this lies in the way NLOS/1 associates one subject to another in its deductive reasoning processes. This will be discussed further when I explain the problem solving algorithm.

Modifiers are adjectives and adverbs. As adjectives, modifiers appearing immediately before a subject relate to that subject. These show what ("red", "lazy", etc.), association ("a", "the", "their", etc. - associative modifiers are generally for aesthetic use only and are ignored by the system in its deductive reasoning), multiplication ("times as many", etc.) and numeric ("100", etc.). Multiplicative and numeric modifiers are very important in the deductive reasoning processes but numbers must appear before multiplicative modifiers in subject clauses (as in "10 times as many boxes", etc.). As adverbs, modifiers may appear before or after a verb phrase and show how ("quickly", etc.).

With all of the above taken into consideration, you can see how the system can break down a sentence into who, what, when, where, why and how catagories of information.

Simple sentences used with the system should have only one piece of information in each of these catagories.

Question invokers may be used in a sentence to request the missing catagory of information represented by the question invoker. For example - if you tell the system "Tom ran home to have dinner", you could later ask "who ran home to have dinner?" and the system would reply "Tom"; or "Tom ran where" would reply "home"; or "why did Tom run home" would reply "to have dinner" assuming that "ran" and "did run" have the same root verb - this will be discussed later.

Question invokers, therefore, can request who, what, when, where, why or how. A special class of question invoker "how many" exists which is used to invoke the deductive reasoning process.

Also, a sentence beginning with a verb phrase invokes a yes/no type of response (given "Tom shined his shoes" you could ask "did Tom shine his shoes?"). Notice that the verb phrase "did shine" has been broken up around the subject to do this. The system would respond "yes" if the information was true or "no" if the information was false or unknown.

Punctuation is not allowed within a defined phrase and is ignored in sentences. Also, numeric modifiers are recognized without formal definition and are not allowed within defined phrases. This would disallow contractions or non-integer numeric modifiers in sentences.

By means of a utility routine in the system, phrases are formally defined to and maintained by the system in a dictionary which stores the phrase, its grammatical usage and the type of information it conveys.

A phrase already in the dictionary may be redefined. To have a phrase deleted or ignored, you simply redefine it as an associative modifier since this type of phrase is for aesthetic purposes only and is ignored in sentence analysis. If a phrase is redefined, the redefinition only applies to sentences analyzed after the redefinition.

Different phrases may contain the same word or words but in different sequences or in combination with other words.

For verb phrases, the system also requires a root phrase which normalizes the time references implied by verbs. For example - "went" and "did go" are both "past go". Whenever a verb phrase is encountered in sentence analysis, it is replaced by its root phrase. This allows such analysis as "Tom went home" and "did Tom go home" with the system replying "yes".

An additional utility lists the vocabulary of phrases along with their characteristics to allow the user to review his dictionary.

As sentences are received and analyzed by the system, the information extracted is stored in a special section of the dictionary structured like an encyclopedia. This section is referenced by use of question invokers or yes/no question invokers as discussed earlier.

A special note - a simple form of compound sentences can be handled by the system during sentence analysis.

A noun followed by a verb may introduce another sentence (as in "Tom said Dick went home"). In this example the system treats the one sentence as two statements: "Tom said what (Dick went home)" and "Dick went home".

The system also contains a utility routine for scrolling thru the encyclopedia section of the dictionary to allow the user to view the information the system has thus far received, analyzed and cataloged. This display shows the information in the sentences and the "information type" catagories they were assigned. The sentences are shown in the sequence they were received to show the flow of changes in facts.

Another function of sentence analysis, which is critical to the deductive reasoning process, is the association of one subject in a sentence relative to the other subjects in the sentence.

For example - in "Tom has 3 balls in a box", two relationships are extracted: "Tom" with "3 balls" and "3 balls" with "box". If you were to ask "how many balls has Tom", the system would reply "3". Notice that the "how many" questioning sentence can relate only one subject to another and can not be conditioned by any how, why or when type clauses.

If a conjunction is used as in "Tom and Dick have a car", the relationships extracted are: "Tom" with "car" and "Dick" with "car"; but, in "Tom and Dick have a car and a truck", if two conjunctions were allowed, the relationships would be "Tom" with "car", "Dick" with "car", "Tom" with "truck" and "Dick" with "truck".

The system can only handle up to 3 subjects in a relationship at one time; so, only one conjunction is allowed per sentence.

These relationships are also stored in a special section of the dictionary structured to handle one to one associations. Along with each of the two subjects in a relationship, the dictionary also stores the numeric modifiers, if any, and the multiplicative modifiers, if any, associated with those subjects. Both subjects in a relationship may not contain multiplicative modifiers and association dictionary entries are not built if the sentence has how, why or when type clauses because the system can not handle conditional relationships in its deductive reasoning processes. For example - in "Tom ran quickly to the store" would not relate "Tom" with "store" because of the how "quickly". Also, "3 times as many balls are used for 2 times the boxes" because of the abstract nature of the relationship between balls and boxes.

And now for the deductive reasoning process - let's consider the following sample problem: given the following dictionary of phrases - a when preposition "on", an associative modifier "their", a what subject "vacation", a who subject "Tom", a who subject "Dick", a where verb "visited" with root "past visit", an associative modifier "a", a where subject "farm", a when preposition "while", a where subject "there", a who subject "he", a what verb "noticed" with root "past notice", a what subject "pen", a what verb "containing" with root "present contain", a what subject "chickens", a what subject "pigs", a what verb "said" with root "past say", a who subject "they", a what verb "were" with root "past be", a multiplicative modifier "times as many", a what preposition "as", a what verb "counted" with root "past count", a what subject "legs", a where preposition "in", an associative modifier "the", a what verb "have" with root "present have", a conjunction "and", and a how many question invoker "how many"; the following sentences can be analyzed by the system - "chickens have 2 legs", "pigs have 4 legs", "on their vacation Tom and Dick visited a farm", "while there they noticed a pen containing chickens and pigs", "Tom said there were 3 times as many chickens as pigs", "Dick said he counted 100 legs in the pen" and "how many chickens were in the pen". The system would reply to the last questioning sentence - "30".

How could the system have deduced this? Let's look at the approach the system takes: the system would first make all the subject to subject relationships possible - "chickens" with "2 legs", "pigs" with "4 legs", "pen" with "chickens", "pen" with "pigs", "Tom" with "there", "there" with "3 times as many chickens", "3 times as many chickens" with "pigs", "Dick" with "he", "he" with "100 legs" and "100 legs" with "pen". Since "pen" is the area of concern in which to search for "how many", the relationships involving "pen" are selected - "chickens", "pigs", "100 legs".

If a non-multiplicative numeric modifier had been associated with chickens, that number would be the answer (the system accepts facts given it as truth; but, if the system had been asked "how many chickens were in 3 pens", the answer found would be multiplied by 3). The system must now look for a relationship with a non-multiplicative numeric modifier to give "pen" a numerical base of equivalence. This relationship would be "100 legs". Since "100 legs" is to be treated as "pen", the remaining "pen" relationships are matched against all of the other relationships in the dictionary in order to pick out these relationships with "legs". This would result in the selection of the following relationships -

"chickens" with "2 legs" and "pigs" with "4 legs". The system must also check for any relationships between the resultant subjects. This search shows the relationship of "3 times as many chickens" with "pigs".

With all of this selected information, an equation can be formulated - "100 legs" equals "3 times as many chickens" plus "pigs". Substituting the number of legs represented by chickens and pigs, a common unit of measure can be applied to the equation. This yields the equation - "100 legs" equals "3 times as many" times "2 legs" plus "4 legs" or 100 equals  $6X + 4X$ . Then X equals 10 and since there are "3 times as many chickens", there must be 3 times 10 or 30 chickens. At this point, the system would simply print "30" and ask for another sentence.

The above algorithm can be applied to a wide variety of math reading style problems. It can be an educational experience just exploring the many variations possible.

The dictionary you construct and its component phrases, sentences and subject relationships can be stored as a file on cassette tape and then reloaded at the beginning of a session. This allows an accumulated data base of information to be maintained for inquiry purposes.

Operationally, NLOS/1, a 16K Level 2 BASIC program, is divided into sections which flow one to another. To begin with, the program asks if you want it to read a dictionary tape created in a previous run. This allows you to build and retain dictionary tapes on various problems and subjects and to re-use this information as needed by having the system re-input the information from tape in a later run. Enter "yes" if you wish a previously created dictionary tape to be read or enter "no" to go on.

After you have entered "no" or the dictionary tape has been read, the system will ask if you are defining a phrase. This section of the program allows the user to add phrase definitions to the system dictionary in memory or to redefine a phrase previously known to the system.

After the phrase definition is entered, this section of the program will repeat until a "no" response is entered to the "DEFINING PHRASES" question. If you enter "yes" to the "DEFINING PHRASES" prompt, you will be expected to answer a series of questions which will supply the information needed to define a phrase to the system.

The first thing the system must know is the phrase text itself. Next, it must be told if the phrase is a 1) subject, 2) verb, 3) preposition, 4) conjunction, 5) modifier, or 6) question invoker by entering the appropriate code.

Once this grammatical usage information is known about a phrase, the system will ask for the type of information that the phrase will convey when used in a sentence. You must enter the proper code from the following list: 1) who, 2) what, 3) when, 4) where, 5) how, 6) how many, 7) association, 8) multiplicative, 9) why.

As discussed previously, the allowable type of information conveyed by a phrase is determined by the grammatical usage. If an improper type of information code is entered, the type of information question will be repeated.

If the phrase being defined is a verb you must next enter the verb phrase which must have been defined which represents the root verb phrase. As discussed previously, this is required to commonize the various phrases which have the same functional usage.

If you wish to delete a phrase from the system's dictionary, simply redefine it with a modifier grammatical usage and an association information type. This will cause the phrase to be ignored during sentence analysis since this class of phrases normally includes words such as "the" or "an" which are functionally unneeded in a sentence.

Once the user has defined all of the phrases needed, you will be given the opportunity to list all the phrases in the dictionary as a review. This will happen if you enter "yes" to "WANT VOCABULARY LIST". If you enter "no" the program will go on.

After this you will be given the opportunity to review the sentences previously inputted if a dictionary tape was loaded by entering "yes" to the "LIST ENCYCLOPEDIA" prompt. These sentences will be displayed in a form broken down into its information type components.

After all of this maintenance and review activity has taken place, the user may begin entering sentences conveying various information by combining phrases previously defined to the system using simple English grammar and enter questions made up of previously defined phrases which ask about information conveyed in previously entered sentences or calculate unknown numerical statistics from quantitative information and relationships conveyed in previously inputted sentences.

The system recognizes a sentence as a question if it begins with a question invoker phrase or if it begins with a verb phrase, in which case the question is taken to be a yes/no confirmation of an informational sentence previously inputted.

Each sentence or question must be entered one at a time in response to the "ENTER SENTENCE OR HIT ENTER". If you hit enter only in response to this prompt, the program will go on to the last section of the system.

During sentence analysis, several error messages may appear. "UNRECOGNIZABLE PHRASE" means that the present sentence being analyzed contains a phrase not in the current dictionary. Any sentences inputted prior to this message are saved in the dictionary. The user can hit the BREAK key and type in "GOTO 455" (ENTER) to put the program back into the "DEFINING PHRASES" section so that the missing phrase can be defined. After doing this and going through the program until you get back to the "ENTER SENTENCE" prompt, you may re-enter the sentence which caused the original error message.

The "INVALID GRAMMAR" message occurs if the sentence being analyzed is too complex for the system. The user should re-enter the same information but using one or more simpler sentences which collectively express the same thought.

"UNKNOWN" and "TOO COMPLEX" messages occur when a "how many" question is too difficult or insufficient data exists in the dictionary. Additional informational sentences may be needed to be inputted before the question is retried or the question might need to be re-expressed in a different form that the system may

understand better.

Finally, the last section of the program writes the dictionary of phrase definitions and sentences to cassette tape if the user responds "yes" to the "SAVE DICTIONARY ON TAPE" prompt. If you respond "no" or after the tape is written, the program ends.

It should be noted that the format of the dictionary uses all string storage. Because of this, the system does not allow punctuation in sentences and numeric modifiers involved in subject associations may only contain non-negative integer numbers. Also, sentence analysis may require several minutes, especially involving "how many" question computation.

In contrast, let us look at the improvements found in NLOS/2 not found in NLOS/1. First, NLOS/2 is a larger, more involved program requiring 32K. The dictionary format has been changed to allow faster execution and more conservation in storage. One drawback to this is that NLOS/2 can not read an NLOS/1 dictionary tape.

Numeric modifiers may contain decimal points and negative values to allow for greater computational flexibility.

Rather than sequentially running through each functional section of the program as we did in NLOS/1, NLOS/2 allows the user to select the section desired by use of a series of commands. Rather than using sectional (yes/no availability) prompts, NLOS/2 uses a general command prompt of a right arrow. The response to this prompt should be "DEFINITION" or "DEF" to initiate a phrase definition, "VOCABULARY" to list the phrase definition portion of the dictionary, "SENTENCE" to list the sentences currently stored in the dictionary, "END" to write the dictionary to tape and end the program, and "PROCEDURE" to input or update the subroutine associated with an action verb phrase.

If the text entered at the arrow prompt is other than one of the above commands, it is assumed to be a sentence or question to be analyzed.

Concerning action verb subroutines maintained by the procedure command, if a question starts with a verb phrase with a subroutine, the sentence is passed to the subroutine and the subroutine is executed as if it were an order or command with the remainder of the sentence treated as a parameter to be interrogated by the subroutine, rather than the sentence being treated as a yes/no question as is implied in a verb phrase with no subroutine associated with it.

As you can see, NLOS/2 has some large advantages as compared to NLOS/1. Some of the other smaller changes include the ability to configure the maximum number of phrase definitions, sentence storage and subject association entries allowed in the dictionary as opposed to NLOS/1 which has a fixed limit on the size of the dictionary; and an instructional subroutine which teaches the user how to set up the coding structure needed for an action verb phrase subroutine to function.

If you would like to run NLOS/2 in only 16K, you would have to give up the action verb subroutine capability. This can be done by using the following changes:

DELETE 1600-1945  
DELETE 7500-7585  
DELETE 9800-9974

1600 RETURN  
7500 RETURN  
9800 GOTO 1140

This would give you a super NLOS/1 with the storage and execution advantages of NLOS/2.

And now, here is the programming listing for NLOS/2:

```

1 CLS:PRINTCHR$(20):PRINTTAB(14)"WLOS":PRINT:PRINTTAB(15)"A":PRINTTAB(8)"NATURAL
LANGUAGE":PRINTTAB(8)"OPERATING SYSTEM":PRINT
2 PRINTTAB(9)"COPYRIGHT 1979":PRINTTAB(10)"CYBERMATE CO.":PRINTTAB(8)"R. D. #3 BOX
192R":PRINTTAB(6)"NR2RARETH":PR 18064":PRINTTAB(7)"PHONE 215-759-6873"
3 PRINT:PRINT" CRSSETTE #4. 95 SOURCE #1. 95":PRINT:INPUT"NEED INSTRUCTIONS(Y/N)":V$:
V$: IFV$="Y":THENGOSUB1600
4 CLS:PRINT"I NEED TO KNOW SOME STATISTICS":DEFINTR:B.C:I
5 INPUT"How much string space may I use":J:CLEARJ:J4+32:IL=4:IK=10
6 INPUT"What is the maximum number of different phrases that you will teach me":I1:
I1: IFI1<10THENI1=10
7 INPUT"What is the maximum number of sentences using those phrases that you wil
l input to inform me on topics":I2: IFI2<10THENI2=10
8 INPUT"What is the maximum number of subject associations I may learn about":I3
:I3: IFI3<10THENI3=10
9 DIMRS(I1),R1(I1,4),R2(I2,8),R3(I3,2),R4(I3,2),R5(I4),R6(I4),R7(I4),R8(I
I3,2),RS(I2)
10 I9=(LEN-256)/5:DIMR7(I9):PRINT"THANK YOU!"
11 FORI6=0TO1:RS(I6)=" ":FORI7=0TO4:R1(I6,I7)=0:NEXTI7:NEXTI6:C1=0
12 FORI6=0TO2:FORI7=0TO8:R2(I6,I7)=" ":NEXTI7:RS(I6)=0:NEXTI6:C2=0
13 FORI6=0TO3:FORI7=0TO2:R3(I6,I7)=" ":R4(I6,I7)=0:R8(I6,I7)=0:NEXTI7:NEXTI6:
C3=0
14 FORI6=0TO4:R5(I6)=0:R6(I6)=" ":NEXTI6
15 FORI6=0TO9:R7(I6)=1:NEXTI6:C3=0
16 FORI6=0TOIK:FORI7=0TOIL:H(I6,I7)=0:NEXTI7:NEXTI6:V2=" "
17 PRINT"You have a total of ";I9;" instruction spaces for procedures!"
25 INPUT"Should I load a data base tape(Y/N)":V$:IFV$<>"Y":THEN140
30 INPUT#1,C1,C2,C3,BH
40 IFC1<1THEN50
45 FORI6=1TOC1:INPUT#1,R0$(I6),R1(I6,1),R1(I6,2),R1(I6,3),R1(I6,4):R0$(I6)=" "+
R0$(I6):NEXTI6
50 IFC2<1THEN60
55 FORI6=1TOC2:INPUT#1,R2$(I6,1),R2$(I6,2),R2$(I6,3),R2$(I6,4),R2$(I6,5),R2$(I6,
6),R2$(I6,7),R2$(I6,8),R2$(I6):NEXTI6
56 FORI6=1TOC2:FORI7=1TO8:R2$(I6,I7)=" "+R2$(I6,I7):NEXTI7:NEXTI6
60 IFC3<1THEN70
65 FORI6=1TOC3:INPUT#1,R3$(I6,1),R3$(I6,2),R4(I6,1),R4(I6,2),R8(I6,1),R8(I6,
2):NEXTI6
70 IFB3<1THEN140
72 C9=0
75 FORI6=1TOC9
80 INPUT#1,B1,B2,B3,B4,B5,B6,B7,B8,B9,BB,BC,BD,BE,BF,BG
85 IFB9O-1THENC9=C9+1:R7(C9)=B0
90 IFB1O-1THENC9=C9+1:R7(C9)=B1
95 IFB2O-1THENC9=C9+1:R7(C9)=B2
100 IFB3O-1THENC9=C9+1:R7(C9)=B3
105 IFB4O-1THENC9=C9+1:R7(C9)=B4
110 IFB5O-1THENC9=C9+1:R7(C9)=B5
115 IFB6O-1THENC9=C9+1:R7(C9)=B6
120 IFB7O-1THENC9=C9+1:R7(C9)=B7
125 IFB8O-1THENC9=C9+1:R7(C9)=B8
130 IFB9O-1THENC9=C9+1:R7(C9)=B9
135 IFB1O-1THENC9=C9+1:R7(C9)=BR
140 IFB2O-1THENC9=C9+1:R7(C9)=BB
145 IFB3O-1THENC9=C9+1:R7(C9)=BC
150 IFB4O-1THENC9=C9+1:R7(C9)=BD
155 IFB5O-1THENC9=C9+1:R7(C9)=BE
160 IFB6O-1THENC9=C9+1:R7(C9)=BF
165 IFB7O-1THENC9=C9+1:R7(C9)=BG
170 NEXTBH:GOT01140
200 IFN=2THENB1=R1(I6,4)
205 RETURN
460 MB$=" ":INPUT"PHRASE":MB$:J=LEN(MB$)
469 MR$=" ":"I7=0:BI=0
470 N=0:PRINT"1-SUBJECT, 2-VERB, 3-PREPOSITION, 4-CONJUNCTION, 5-MODIFIER, 6-QUESTION
INVOKE":INPUTN:IFN>10THEN26:IFN<1

```

480 K=7:0NNGOT0498,500,510,500,530,550  
 490 INPUT"1=WHO, 2=WHAT, 4=WHERE"; K: IFK=10RK=20RK=4THEN560ELSE490  
 INPUT"2=WHAT, 4=WHERE"; K: JFk=20RK=4THEN560ELSE500  
 INPUT"2=WHAT, 3=WHEN, 4=WHERE, 5=HOW, 9=WHY"; K: JFk=20RK=30RK=40RK=50RK=9THEN560E  
 LSE510  
 530 INPUT"2=WHAT, 5=HOW, 7=ASSOCIATION, 8=MULTIPLICATIVE"; K: JFk=20RK=50RK=70RK=8THE  
 NS60ELSE530  
 550 INPUT"1=WHO, 2=WHAT, 3=WHEN, 4=WHERE, 5=HOW, 6=HOW MANY, 9=WHY"; K: JFk=10RK=20RK=20  
 RK=40RK=50RK=60RK=9THEN560ELSE550  
 560 IFNO2THEN600  
 575 IFCl(1THEN612  
 580 INPUT"ROOT VERB PHRASE"; M\$  
 582 FOR16=1TO1: IFR0\$(16)=M\$ THEN17=16  
 583 NEXT16  
 590 IFCl(1THEN612  
 610 FOR16=1TO1: IFR0\$(16)=M\$ THEN0SUB200: GOT0620  
 611 NEXT16  
 612 16=C1+1: IF16>11 THENPRINT"TOO MANY PHRASES!": RETURN  
 620 R0\$(16)=M\$: R1(16,1)=N: R1(16,2)=K: R1(16,3)=T: R1(16,4)=B1: C1=16  
 621 IF17=0 THENR1(16,3)=16  
 622 RETURN  
 775 INPUT"PHRASE"; M\$; I6=1: IFM\$="ALL" AND1>0 THEN790  
 776 IFCl(1THENRETURN  
 777 FOR16=1TO1: IFR0\$(16)=M\$ THEN790  
 /NEXT16: RETURN  
 779 IF16>C1 THENRETURN  
 790 N=R1(16,1): K=R1(16,2): T=R1(16,3): B=R1(16,4)  
 800 PRINT"PHRASE-"; R0\$(16)  
 840 IFN=1V\$="SUBJECT"  
 850 IFN=2V\$="VERB"  
 860 IFN=3V\$="PREPOSITION"  
 870 IFN=4V\$="CONJUNCTION"  
 880 IFN=5V\$="MODIFIER"  
 900 IFN=6V\$="QUESTION INVOKER"  
 920 PRINT"GRAMMAR-"; V\$: P=P+1: IFK=1V\$="WHO"  
 940 IFK=2V\$="WHAT"  
 950 IFK=3V\$="WHEN"  
 960 IFK=4V\$="WHERE"  
 970 IFK=5V\$="HOW"  
 980 IFK=6V\$="HOW MANY"  
 990 IFK=7V\$="ASSOCIATION"  
 1000 IFK=8V\$="MULTIPLICATIVE"  
 1015 IFK=9V\$="WHY"  
 1020 PRINT"USAGE-"; V\$: IFN=2THENPRINT"ROOT VERB PHRASE-"; R0\$(17)  
 1028 IFM\$="ALL" THENINPUT"Hit ENTER TO CONTINUE"; V\$: I6=16+1: GOT0779

1029 RETURN  
 1140 V\$=" "; INPUT">"; V\$: P=0  
 1141 IFV\$="PROCEDURE" THEN0SUB7500: GOT01140  
 1142 IFV\$="DEFINITION" THEN0SUB460: GOT01140  
 1143 IFV\$="VOCABULARY" THEN0SUB775: GOT01140  
 1144 IFV\$="SENTENCE" THEN0SUB8500: GOT01140  
 1145 IFV\$="END" ORV\$=" " ORV\$=" THEN8000  
 1146 IFV\$="END" ORV\$=" " ORV\$=" THEN8000  
 1149 V\$=V\$+" "  
 1150 J=LEN(V\$): ZB=0: T=0: ZB\$=" ": B\$=" "  
 1151 P1=0: P2=0  
 1160 ZH=0: ZI=0: ZD=0: N=0: K=0: L=0: ZE=0: XH=0: ZZ=0  
 1165 M1\$=" "; M2\$=" "; M3\$=" "; M4\$=" "; M5\$=" "; M6\$=" "; M7\$=" "; E\$=" "; ZA=0: B\$=" "  
 ZY=0: Q\$=" "  
 1167 S1\$=" "; S2\$=" "; S3\$=" "; S4\$=" "; V1=0: V2=0: V3=0: V4=0: X1=0: X2=0: X3=0: X4=0: D\$=0  
 "  
 1168 SR=0  
 1170 P=P+1: IFP>JTHEN8000  
 1180 F1=RSD(MID\$(V\$, P, 1)): IFF1=32THEN1170  
 1181 IFSR=0 THENSR=P  
 1182 IFF1=420RF1=450RF1=46THEN1195  
 1190 IFF1=480RF1=577THEN1240  
 1196 SL=1  
 1198 SR=P  
 1200 P=P+1: IFP>JTHEN1230  
 1200 F1=RSD(MID\$(V\$, P, 1)): IFF1=32THEN1230  
 1221 SL=SL+1: GOT01200  
 1230 CS=MID\$(V\$, SR, SL)  
 1231 T=VAL(C\$): N=5: K=10  
 1232 SR=0: SR=1: SL=1  
 1233 GOT01340  
 1240 SR=-1: SL=0: IFCl(1THEN1250  
 1241 FOR16=1TO1  
 1242 IFLEN(V\$)-P+1>LEN(R0\$(16)) THEN1246  
 1243 SB=LEN(R0\$(16))  
 1245 IFR0\$(16)=MID\$(V\$, P, SB)AND50>SL THENSR=16: SL=SB  
 1246 NEXT16  
 1250 IFSR=-1 THEN2222  
 1260 CF=R0\$(SR): N=R1(SR, 1): K=R1(SR, 2): S9=SR: SR=SL  
 1340 IFN=1AND2AND2B=1THEN2222  
 1341 IF2H=3ANDNC4THENZH=0: GOTSUB3970  
 1342 IFN=4AND2B=1THEN2222  
 1343 IF2H=3OR2H=9THEN4100  
 1344 IFN=4ANDZB=1THEN2222  
 1345 IFN=4ANDZB=7THEN2222

1346 IFZH=3THEN04=0\$+" "+C\$  
 1347 IFZB=4ANDNO1THEN2222  
 1348 IFN=5ANDC7THEN04=0\$+" "+C\$  
 1349 IFN=5B\$+0\$+" "+C\$  
 1350 IFN=3THEN04=0\$;ZH=3;21-K  
 1351 IFZH=3THEN04=17H-9  
 1352 IFN=6THEN2000  
 1353 IFN=5ANDK=3M=1  
 1354 IFN=5ANDK=8AND12THEN2222  
 1355 IFN=2AND2A=6AND2B=820=8  
 1356 IFZB=8ANDR1(S9,4)D01HEN3900  
 1357 IFN=3ANDZH=2AND0\$/" "M64-M64"/"10\$;0\$+" "+B\$+" "  
 1358 IFN=11HEN0\$=0\$+" "+C\$;0\$+" "  
 1359 IFN=1RN08\$C+" 0\$-B\$+" "+C\$;0\$+" "+B\$+" "  
 1360 IF2NC3NDN=100SUE4300  
 1361 IFN=2RN02\$C+" M6\$-M6\$+" /"10\$;0\$+" "+B\$+" "  
 1362 IFN=428=1  
 1363 IFN=2RN02B=8ANDP120THEN203=MID\$(V\$,P1):22=P2:GOSUB213:P=P1:G0101150  
 1364 IFN=200SUE3100  
 1365 IFN=1RN02B=300SUE4000  
 1366 IFN=1RN02B=200SUE4800  
 1367 IFN=1RN02B=400SUE4800  
 1368 IFN=11=0:XM=0:2Y=2H4  
 1369 G0102600  
 1370 PRINT"NO DS/2 USERS MUST HAVE A WORKING KNOWLEDGE OF M65/1!"  
 1371 PRINT"WHEN THE > PROMPT APPEARS, YOU MUST ENTER A COMMAND OR"  
 1372 PRINT"A SENTENCE OR QUESTION. THE COMMANDS ARE AS FOLLOWS -"  
 1373 PRINT"DEFINITION - ALLOWS YOU TO DEFINE A PHRASE TO THE SYSTEM"  
 1374 PRINT"VOCABULARY - ALLOWS YOU TO EXAMINE A PHRASE DEFINITION OR"  
 1375 PRINT"ENTER ALL TO EXAMINE ALL PHRASES. SENTENCE - ALLOWS YOU TO"  
 1376 PRINT"EXAMINE SENTENCES PREVIOUSLY INPUTTED. PROCEDURE - ALLOWS"  
 1377 PRINT"YOU TO DEFINE A BASIC PROGRAM TO BE RUN WHEN THE SYSTEM"  
 1378 PRINT"SEES A PARTICULAR VERB PHRASE. THIS INVOLVES ENTERING A"  
 1379 PRINT"SERIES OF OP CODES AND THEIR OPERANDS WHICH THE SYSTEM WILL"  
 1380 PRINT"TRANSLATE INTO A BASIC PROGRAM. THE SYSTEM SUPPLIES WORK"  
 1381 PRINT"SPACE FOR YOUR BASIC ROUTINE IN THE FORM OF 33 SINGLE"  
 1382 PRINT"PRECISION FIELDS NUMBERED 0 THRU 32 AND 33 STRING FIELDS"  
 1383 PRINT"NUMBERED 0 THRU 32. THE WORK SPACE IS INITIALIZED PRIOR"  
 1384 PRINT"TO RUNNING THE VERB PROCEDURE EXCEPT FOR STRING 0 WHICH"  
 1385 INPUT"Hit ENTER TO CONTINUE":V\$:CLS  
 1386 PRINT"CONTAINS THE SENTENCE CONTAINING THE VERB PHRASE JUST"  
 1387 PRINT"INPUTTED. THE VERB PROCEDURE IS INPUTTED INTO A BUFFER"  
 1388 PRINT"ALLOCATED FROM CONTIGUOUS INTEGER FIELDS. YOU MUST TELL"  
 1389 PRINT"THE SYSTEM HOW MANY INTEGER SPACES TO ALLOW FOR A VERB."  
 1390 PRINT" EACH SPACE IS ADDRESSED NUMERICALLY FROM 1 TO N. FOR A NEW"  
 1391 PRINT"PROCEDURE, ALL SPACES ARE INITIALIZED TO 0. SPACES MAY BE"  
 1392 PRINT"LISTED, CHANGED IN VALUE OR HAVE THE RSC VALUES OF R"  
 1393 PRINT"STRING INSERTED STARTING AT A SPECIFIED LOCATION WITH"  
 1394 PRINT"THE STRING LENGTH INSERTED BEFORE THE STRING. THIS IS"  
 1395 PRINT"USED TO INPUT STRING OPERANDS FOR OP CODES THAT REQUIRE"  
 1396 PRINT"THEM. ANY OPERANDS REQUIRED BY AN OP CODE MUST FOLLOW"  
 1397 PRINT"THAT OP CODE IN THE INSTRUCTION SPACE. HERE ARE THE OP"  
 1398 PRINT"CODES AND THEIR OPERAND REQUIREMENTS -"  
 1399 INPUT"Hit ENTER TO CONTINUE":V\$:CLS  
 1400 PRINT"OP CODE NAME, OPERANDS (S-STRING, N-NUMERIC FIELD, R-STRING"  
 1401 PRINT"1-FIELD, 1-NUMERIC STRING, L-LABEL NUMBER)"  
 1402 PRINT"0-NOP 1-CLS 2-LET,R,S 3-LET,N,T 4-IF,R,R 5-IF,N,N 6-POKE,N,N"  
 1403 PRINT"7-PEEK,N,N 8-RDN,N,N 9-SUB16621,N,N 10-MULTIPLY,N,N"  
 1404 PRINT"11-DIVIDE,N,N 12-L,N,N 13-\$+,R,R 14-INT,N,N"  
 1405 PRINT"15-SIN,N,N 16-COS,N,N 17-TAN,N,N 18-RTN,N,N 19-VRL,N,R"  
 1406 PRINT"20-STRL,R,N 21-STRING\$,R,N,R 22-PRINT,R 23-INKEY\$,R"  
 1407 PRINT"24-JUMP TO LABEL,L 25-RES,N,N 26-INPUT,N,N 27-PRINT1,R"  
 1408 PRINT"28-PRINTG,N,R 29-INPUT1,R 30-INPUT1,I,R 31-PRINTH-1,R"  
 1409 PRINT"32-LEFT\$,R,R,N 33-RIGHT\$,R,R,N 34-MID\$,R,R,N,N 35-MOVE,N,N"  
 1410 PRINT"36-MOVE,R,R 37-EXP,N,N 38-LOG,N,N 39-JUMP IF =,L"  
 1411 PRINT"40-JUMP IF >,L 41-JUMP IF <,L 42-RCG,N,R 43-CHR\$,R,N"  
 1412 PRINT"44-LEN,N,R 45-SQR,N,N 46-RETURN 47-PRINTN,N 48-SET,I,N,N"  
 1413 PRINT"49-RESET,N,N 50-POINT,N,N,N 51-OUT,N,N 52-RND(0),N"  
 1414 INPUT"Hit ENTER TO CONTINUE":V\$:CLS  
 1415 PRINT"53-INSTR,R,R,N 54-END 55-GOSUB,L 999-LABEL,L"  
 1416 PRINT"MOST OP CODES AND THEIR OPERANDS OPERATE LIKE THEIR BASIC"  
 1417 PRINT"COUNTERPARTS WITH THE FIRST OPERAND BEING THE RESULT FIELD."  
 1418 PRINT"POINT AND INSTR PUT 1 IN THE RESULT NUMERIC FIELD WITH"  
 1419 PRINT"THE FIRST OPERAND OF INSTR BEING THE LARGER STRING IN THE"  
 1420 PRINT"SEARCH. A JUMP SHOULD FOLLOW AN IF INSTRUCTION."  
 1421 PRINT"YOU SHOULD NOT ATTEMPT TO PROGRAM A VERB PROCEDURE UNLESS"  
 1422 PRINT"YOU ARE EXPERIENCED IN LEVEL 2 BASIC PROGRAMMING!"  
 1423 INPUT"Hit ENTER TO CONTINUE":V\$:CLS:RETURN  
 2000 P=P1\$R-1:IFN=2P1=P1  
 2001 IFN=2B-N:ZE=K  
 2002 IFN=2P2=ZI  
 2010 G0101170  
 2020 IFZB=10ZB040R8\$C+" THEN2222  
 2025 IFK=300=5  
 2030 IFK=170=1  
 2040 IFK=220=2  
 2050 IFK=320=3  
 2060 IFK=420=4  
 2070 IFK=520=6

2000 IFK=620=7  
 2005 IF20=7AND28=1THEN2222  
 J GOT02000  
 J GOSUB3000:C2=C2+1:R2\$(C2,1)=#1\$:R2\$(C2,2)=#2\$:R2\$(C2,3)=#3\$:R2\$(C2,4)=#4\$:R2\$(C2,5)=#5\$:R2\$(C2,6)=#7\$:R2\$(C2,7)=#8\$:R2\$(C2,8)=#9\$:R9(C2)=22:Z2=0:Z0\$="# "  
 2155 M1\$="" ;M2\$="" ;M3\$="" ;M4\$="" ;M5\$="" ;M6\$="" ;M7\$="" :RETURN  
 2222 PRINT" I DO NOT UNDERSTAND THIS SENTENCE!":GOT01140  
 3000 FORI6=1TO01:IFM7\$=R0\$(I6)THENI7=R1(I6,3):M7\$=R0\$(I7)  
 3010 NEXTI6:RETURN  
 3070 IF21=9M5\$=M5\$+"/"1D\$  
 3075 IF21=M6\$=M6\$+"/"1D\$  
 3080 IF21=2M2\$=M2\$+"/"1D\$  
 3090 IF21=3M3\$=M3\$+"/"1D\$  
 3100 IF21=4M4\$=M4\$+"/"1D\$  
 3110 D\$=" " :RETURN  
 3180 IFM17\$=" "THEM17\$=0ELSEM17\$=M7\$+" "+0\$  
 3190 ZH=N:ZI=K:RETURN  
 4100 IFN=10RN=5THEN1344  
 4110 IFZH=39NDN=4ZH=3:GOT01344  
 4120 GOT02222  
 4300 ZP=2RN1:IFK=1M1\$=M1\$+"/"1D\$  
 4320 IFK=2M2\$=M2\$+"/"1D\$  
 4330 IFK=4M4\$=M4\$+"/"1D\$  
 J RETURN  
 J S2\$=S3\$:S2\$=S4\$:W1=V3:V2=V4:X1=X3:X2=X4:S3\$=" "14\$=" " ;V3=0:V4=0:X3=0:X4=0  
 :RETURN  
 4700 S3\$=E\$:V3=T:X3=XN:GOSUB4900:RETURN  
 4800 S4\$=E\$:V4=T:X4=XN:GOSUB4900:RETURN  
 4900 IPS1\$O=" "F\$=S1\$:F1=V1:F2=V1:GOSUB5000  
 4910 IPS2\$O=" "F\$=S2\$:F1=V2:F2=V2:GOSUB5000  
 4920 RETURN  
 5000 JTH64O=" "0RM3\$O" "0RM5\$O" "RETURN  
 5001 JFF2=1ANDXN=1&RETURN  
 5002 IFZ0=0&RETURN  
 5005 C3=C3+1:R2\$(C3,1)=F\$:R3\$(C3,2)=E\$:R4(C3,1)=F2:R4(C3,2)=X\$  
 5010 R8!(C3,1)=F1:R8!(C3,2)=T:RETURN  
 6000 IF0\$O=" "M6\$=M6\$+"/"1D\$  
 6001 IFZH=3THEN0500:GOT02000  
 6002 IFZ0=0THEN6200  
 6005 IFM7\$=" "THEN2222ELSE6005:GOT01140  
 6200 IFZ0=7THEN7000ELSE1:P=0  
 6205 IFC1(L)THEN6210  
 6206 FORI8=1TO01:IFLEN(M7\$)>OLEN(R0\$(I8))ORLEN(I8)>OM7\$THEN6208  
 6207 Z2=R1(I8,3):IFR2>0THENM7\$=R0\$(B2)  
 J NEXTI8  
 6210 L=1:P=0  
 6220 IFC2(L)THEN6200  
 6230 FORIB=1TOC2  
 6300 IFM1\$=" "THEN6320  
 6310 M1\$=R2\$(IB,1):M8\$=#1\$:GOSUB9700:IFM6=0THEN6600  
 6320 IFM2\$=" "THEN6340  
 6330 M1\$=R2\$(IB,2):M8\$=#2\$:GOSUB9700:IFM6=0THEN6600  
 6340 IFM3\$=" "THEN6360  
 6350 M1\$=R2\$(IB,3):M8\$=#3\$:GOSUB9700:IFM6=0THEN6600  
 6360 IFM4\$=" "THEN6380  
 6370 M1\$=R2\$(IB,4):M8\$=#4\$:GOSUB9700:IFM6=0THEN6600  
 6380 IFM5\$=" "THEN6400  
 6390 M1\$=R2\$(IB,7):M8\$=#5\$:GOSUB9700:IFM6=0THEN6600  
 6400 IFM6\$=" "THEN6420  
 6410 M1\$=R2\$(IB,5):M8\$=#6\$:GOSUB9700:IFM6=0THEN6600  
 6420 IFM7\$=R2\$(IB,6)THEN6460ELSE6600  
 6450 M1\$=R2\$(IB,3):M8\$=#0\$:GOSUB9700:IFM6=0THEN6600  
 6460 P=18  
 6500 NEXTIB  
 6700 IFZ0<3ANDP=0THEN7920  
 6705 IFZ0=8ANDP=0PRINT"NO":GOT01140  
 6720 IFZ0=8ANDR2\$(P,8)=0:GOT01140  
 6730 IFZ0=1ANDR2\$(P,1)=0" "THENPRINTR2\$(P,1):GOT01140  
 6740 IFZ0=2ANDR2\$(P,2)=0" "THENPRINTR2\$(P,2):GOT01140  
 6750 IFZ0=3ANDR2\$(P,3)=0" "THENPRINTR2\$(P,3):GOT01140  
 6760 IFZ0=4ANDR2\$(P,4)=0" "THENPRINTR2\$(P,4):GOT01140  
 6770 IFZ0=5ANDR2\$(P,7)=0" "THENPRINTR2\$(P,7):GOT01140  
 6780 IFZ0=6ANDR2\$(P,5)=0" "THENPRINTR2\$(P,5):GOT01140  
 6800 IFZ0=8PRINT"YES":GOT01140  
 6810 GOT07920  
 7000 IFX1=10RV1200RX3=10R2Y20R3ZO" "0RM5\$O" "0RM6\$O" "0RS1\$=" "0RS3\$=" "THE  
 N2222  
 7001 JK=10:IL=4  
 7002 L=1:FORN=1TOJK:FORK=1TOIL:H(N,K)=0:NEXTK:NEXTN:KI=0:N=0:ZD=0  
 7003 IFC3(L)THEN7920  
 7004 L=0  
 7015 GOSUB7900  
 7020 IPSR=9THEN7060  
 7022 P=0:J=0  
 7025 IFR3\$(L,1)=S3\$THENP=2:J=1  
 7026 IFR3\$(L,2)=S3\$THENP=1:J=2  
 7030 IFP=0ORJ=0THEN7015  
 7031 IFR4(L,P)=10RR4(L,J)=1THEN7015  
 7032 IFR8!(L,J)>1THEN7015  
 7035 IKI=0THEN7043

7037 T=0  
 7039 T=T+1:Z2=H(1,1):Z0=H(1,4):IFR3\$(Z0,Z2)=R3\$(L,P)THEN7044  
 7041 IF1\$KITHEN7039  
 7043 K1=K1H1:T=K1  
 7044 IFK1DIK1THEN7036  
 7045 Z9=R3\$(L,P):IFZD00THEND=1  
 7047 H(1,1)=P:H(1,2)=Z9:H(1,4)=L:IFR3\$(L,P)=S1\$THENN=1  
 7049 G01070213  
 7050 IFN=00KI=0THEN7020  
 7052 Z1=N:Z9=H(N,2):IFZD00THEN7020  
 7056 IFZD=0THEN7020  
 7110 FORP=110KI:T=H(P,1):Z0=H(P,4):IFR(P,2)=0THEN7115ELSE7200  
 7115 ZE=0:FORN=110KI:K=H(N,1):K3=H(N,4):IFN=PTHEN7250ELSEH(N,3)=0  
 7130 L=0  
 7149 GSUB7200:Z2=0:Z0=0:IFSR=0THEN7240  
 7150 IFR3\$(Z0,T)=R3\$(L,1)THENZ2=1:ZB=2  
 7170 IFR3\$(Z0,1)=R3\$(L,2)THENZ2=2:ZB=1  
 7180 IFZ2=0THEN7140  
 7190 IFR3\$(L,ZB)=R3\$(K3,K)THEN7200ELSE7140  
 7200 ZD=R3\$(L,ZZ):ZB=R3\$(L,ZB):IFZD00ANDZB=0THEN7230ELSE7140  
 7230 H(N,3)=ZB:G0107140  
 7240 IFH(N,3)202E-ZE+1  
 7250 NEXTN:IFZE=K1-1THEN7270  
 7260 NEXTP:G0107320  
 7270 FORM=1TOKI:IFP=NTHEN7290  
 7280 IFH(N,2)=0THENH(N,2)=H(N,2)\*H(N,3)ELSEH(N,2)=H(N,3)  
 7290 H(N,3)=0:NEXTN  
 7300 FORM=1TOKI:IFN=PTHEN7420ELSE1=H(N,1)  
 7305 ZD=H(N,4)  
 7310 FORK=1TOKI:IFK=PORK-NTHEN7410  
 7320 L=1:Z2=H(K,1)  
 7322 L=0  
 7325 K3=H(K,4)  
 7330 GSUB7200:ZC=0:ZD=0  
 7340 IFSR=0THEN7410  
 7350 IFR3\$(ZB,T)=R3\$(L,1)THENZC=1:ZD=2  
 7360 IFR3\$(ZB,1)=R3\$(L,2)THENZC=2:ZD=1  
 7370 IFZD=0THEN7330  
 7380 IFR3\$(K3,2Z)=R3\$(L,2D)THEN7390ELSE7300  
 7390 IFR4(L,1)=100R4(L,2)=1THEN7400ELSE7300  
 7400 H(N,3)=R8\$(L,2D):H(K,3)=R8\$(L,2D):G0107320  
 7410 NEXTK  
 7420 NEXTN  
 7430 FORM=1TOKI:IFN=PTHEN7450  
 7440 IFH(N,3)20H(N,2)=H(N,2)\*H(N,3)  
 7450 NEXTN  
 7460 T=0:FORN=1TOKI:IFN0PTHENT=T+H(N,2):NEXTN  
 7470 ZB=H(P,2)/T:IFH(21,3)=0THENZB=ZB/H(21,3)  
 7480 G0107320  
 7490 IFDC1C1THENRETURN  
 7500 M8\$=""":P=0:Q=0  
 7505 INPUT"VERB PHRASE":M8\$  
 7510 FORI6=110C1:IFR8\$(16)=M8\$THEN?520  
 7515 NEXTI6:PRINT"UNKNOWN":RETURN  
 7520 I7=I1(16,4):IFI700THEN?540  
 7521 IFI1(16,1)Q2THENPRINT"NOT A VERB":RETURN  
 7522 PRINT"NEW ACTION VERB PROC"  
 7525 B9=C9+1:IFB9>1219THENPRINT"OUT OF MEMORY":END  
 7530 C9=B9:I7=B9:I1(16,4)=B9  
 7532 B4=0:INPUT"# INSTRUCTION SPACES":B4:IFB4<219THEN742  
 7533 IFC9+B4-1219THENPRINT"INSUFFICIENT MEMORY":G0107532  
 7534 R7(I7)=B4-1:B6=17:FORS=110B4-1:B6=B6+1:R7(B6)=0:NEXTBS:PRINT"INITIALIZED 1  
 0 NOP\\$"  
 7535 C9=C9+B4-1  
 7540 INPUT"L=LIST, C=CHANGE, S=STRING, E=END":M8\$  
 7545 IFM8\$~"E"THENRETURN  
 7546 IFM8\$~"S"THEN7520  
 7550 INPUT"FROM TO LOCATIONS":B1,B2:IFB1>80ORB1<10RS2>R7(I7)THEN7550  
 7553 IFM8\$~"C"THEN7560  
 7555 FORS=81TOB2:PRINT" LOC ">B3;" CONTENTS ">R7(I7+B3);  
 7560 IFR7(I7+B3)>80ANDR7(I7+B3)<255THENPRINT" RSC ">CHR\$(R7(I7+B3))ELSEPRINT" "  
 7567 NEXTB3:G0107540  
 7565 FORS=81TOB2:PRINT" LOC ">B3;" MRS ">R7(I7+B3);  
 7566 IFR7(I7+B3)>80ANDR7(I7+B3)<255THENPRINT" RSC ">CHR\$(R7(I7+B3))ELSEPRINT" "  
 7567 INPUT"NOW":R7(I7+B3)  
 7570 NEXTB3:G0107540  
 7575 INPUT"LENGTH POINTER LOCATION":B1:IFB1<10RS1>R7(I7)THEN7575  
 7580 M8\$="":INPUT"STRING VALUE":M8\$:S2=LEN(M8\$):IFB2<10RS2>R7(I7)THEN7580  
 7582 R7(I7+B1)=S2  
 7585 FORS=110R2:R7(I7+B1+B3)=RSC(MID\$(M8\$,S2,1)):NEXTB3:G0107540  
 7590 L=L+1:IFL>C3THENSR=9:L=0ELSESR=0  
 7615 RETURN  
 7620 PRINT"UNKNOWN":G0101140  
 7625 PRINT"100 COMPLEX":G0101140  
 7630 IFV2028R=ZB#V3  
 7640 PRINT2R:G0101140  
 8000 IFC1200RC2000RC3200RC920THEN800ELSESEND  
 8005 INPUT"STORE DATA BASE ON TAPE(Y/N)":V\$  
 8010 IFV\$~"Y"THENEND  
 8021 B=INT(C9/I7):H

```

8022 PRINT#-1, CL, C2, C3, BH
8025 IFC1(1THEN8035
8026 FORI6=1TOC1
8027 PRINT#-1, A$$(I6), R1(I6,1), R1(I6,2), R1(I6,3), R1(I6,4)
8033 NEXTI6
8035 IFC2(1THEN8045
8046 FORI6=1TOC2
8042 PRINT#-1, R2$(I6,1), R2$(I6,2), R2$(I6,3), R2$(I6,4), R2$(I6,5), R2$(I6,6), R2$(I6,7), R2$(I6,8), R8(I6)
8043 NEXTI6
8045 IFC3(1THEN8055
8050 FORI6=1TOC3
8052 PRINT#-1, R3$(I6,1), R3$(I6,2), R4(I6,1), R4(I6,2), R8!(I6,1), R8!(I6,2)
8053 NEXTI6
8055 IFBK1THENEND
8060 FORI6=1TOBH
8062 I7=0
8065 B8=-1: I7=I7+1: IFI7<=C9THENB9=R7(I7)
8070 B1=-1: I7=I7+1: IFI7<=C9THENB1=R7(I7)
8075 B2=-1: I7=I7+1: IFI7<=C9THENB2=R7(I7)
8080 B3=-1: I7=I7+1: IFI7<=C9THENB3=R7(I7)
8085 B4=-1: I7=I7+1: IFI7<=C9THENB4=R7(I7)
8090 B5=-1: I7=I7+1: IFI7<=C9THENB5=R7(I7)
8095 B6=-1: I7=I7+1: IFI7<=C9THENB6=R7(I7)
8100 B7=-1: I7=I7+1: IFI7<=C9THENB7=R7(I7)
8105 B8=-1: I7=I7+1: IFI7<=C9THENB8=R7(I7)
8110 B9=-1: I7=I7+1: IFI7<=C9THENB9=R7(I7)
8115 B0=-1: I7=I7+1: IFI7<=C9THENB0=R7(I7)
8120 B1=-1: I7=I7+1: IFI7<=C9THENB1=R7(I7)
8125 B2=-1: I7=I7+1: IFI7<=C9THENB2=R7(I7)
8130 B3=-1: I7=I7+1: IFI7<=C9THENB3=R7(I7)
8135 B4=-1: I7=I7+1: IFI7<=C9THENB4=R7(I7)
8140 B5=-1: I7=I7+1: IFI7<=C9THENB5=R7(I7)
8145 B6=-1: I7=I7+1: IFI7<=C9THENB6=R7(I7)
8150 PRINT#-1, B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, BB, BC, BD, BE, BF, BG
8155 NEXTI6: END
8500 IFC2(1THENRETURN
8502 FORI6=1TOC2:CLS:PRINT"WHEN-";R2$(I6,3):PRINT"WHY-";R2$(I6,7)
8505 PRINT"WHO-";R2$(I6,1):PRINT"WHAT-";R2$(I6,2):PRINT"WHERE-";R2$(I6,4)
8510 PRINT"VERB-";R2$(I6,6):PRINT"How-";R2$(I6,5):PRINT"OBJ CLAUSE-";R2$(I6,8)
8515 INPUT"Hit ENTER TO CONTINUE":V$
8520 NEXTI6:RETURN
8700 MB=0:IFLEN(MB$)<LEN(MB$)RETURN
8705 IPLEFT$(MB$,2)=" /":MB$=MID$(MB$,3,LEN(MB$)-2)
8710 FORMF=1TOLEN(MB$)-LEN(MB$)+1
8720 IFMB$=MID$(MB$,MF,LEN(MB$))MF=1:RETURN
8730 NEXTMF:RETURN
8800 FORL=0TO14:R6$(L)=":":R5!(L)=0:NEXTL:R6$(0)=V$:GT=0:LT=0:ED=0:RT=0
8801 I7=R1(S9,4):P=I7
8802 I6=R7(I7)
8803 I7=I7+1:L=R7(I7)
8804 IFL=0THEN8003
8806 IFL=1THENOL5:GOT08003
8807 IFL=2THENGOSUB9970:V$="":FORK=1T05R:I7=I7+1:N=R7(I7):V$=V$+CHR$(N):NEXTK:R5$(L)=V$:GOT08003
8808 IFL=3THENGOSUB9970:V$="":FORK=1T05R:I7=I7+1:N=R7(I7):V$=V$+CHR$(N):NEXTK:R5$(L)=V$:GOT08003
8809 IFL=4THEN8015
8810 GOSUB9970:GT=0:LT=0:ED=0
8811 IFR6$(L)=R6$(SR)THENED=1
8812 IFR6$(L)>R6$(SR)THENGT=1
8813 IFR6$(L)<R6$(SR)THENLT=1
8814 GOT08003
8815 IFLO4THEN8021
8816 GOSUB9970:GT=0:LT=0:ED=0
8817 IFR5!(L)=R5!(SR)THENED=1
8818 IFR5!(L)>R5!(SR)THENGT=1
8819 IFR5!(L)<R5!(SR)THENLT=1
8820 GOT08003
8821 IFL=6THENGOSUB9970:N=R5!(L):K=R5!(SR):POKEK,K:GOT08003
8822 IFL=7THENGOSUB9970:R5!(L)=PEEK(R5!(SR)):GOT08003
8823 IFL=8THENGOSUB9970:R5!(L)=R5!(L)+R5!(SR):GOT08003
8824 IFL=9THENGOSUB9970:R5!(L)=R5!(L)-R5!(SR):GOT08003
8825 IFL=10THENGOSUB9970:R5!(L)=R5!(L)+R5!(SR):GOT08003
8826 IFL=11THENGOSUB9970:R5!(L)=R5!(L)/R5!(SR):GOT08003
8827 IFL=12THENGOSUB9970:R5!(L)=R5!(L)\R5!(SR):GOT08003
8828 IFL=13THENGOSUB9970:R6$(L)=R6$(L)+R6$(SR):GOT08003
8829 IFL=14THENGOSUB9970:N=INT(R5!(SR)):R5!(L)=N:GOT08003
8830 IFL=15THENGOSUB9970:R5!(L)=SIN(R5!(SR)):GOT08003
8831 IFL=16THENGOSUB9970:R5!(L)=COS(R5!(SR)):GOT08003
8832 IFL=17THENGOSUB9970:R5!(L)=TAN(R5!(SR)):GOT08003
8833 IFL=18THENGOSUB9970:R5!(L)=ATN(R5!(SR)):GOT08003
8834 IFL=19THENGOSUB9970:R5!(L)=VRL(R6$(SR)):GOT08003
8835 IFL=20THENGOSUB9970:R6$(L)=STR$(R5!(SR)):GOT08003
8836 IFL=21THENGOSUB9968:R6$(K)=STRING$(R5!(L),R6$(SR)):GOT08003
8837 IFL=22THENGOSUB9972:PRINTA$$(SR):GOT08003
8838 IFL=23THENGOSUB9972:R6$(SR)=INKEY$:GOT08003
8839 IFL=24THEN8045
8840 GOSUB9972:I7=P:FORN=1TO16
8841 I7=I7+1:IFR7(I7)<999THEN8044

```

```

9842 I7=I7+1:IF A7(I7)<>R7 THEN GOTO 9844
9843 GOT09803
9844 NEXTN:PRINT"INVALID TAG":SR:GOTD1140
9845 IFL=25THEN GO SUB 9970:RS!(L)=ABS(RS!(SR)):GOTD9803
9846 IFL=26THEN GO SUB 9970:RS!(L)=INV(RS!(SR)):GOTD9803
9847 IFL=27THEN GO SUB 9972:PRINT A$(SR):GOTD9803
9848 IFL=28THEN GO SUB 9970:PRINT RS!(L),RS!(SR):GOTD9803
9849 IFL=29THEN GO SUB 9972:INPUT A$(SR):GOTD9803
9850 IFL=30THEN GO SUB 9972:INPUT#1,A$:RS!:GOTD9803
9851 IFL=31THEN GO SUB 9972:PRINT#1,CHR$(34)+RS!(SR)+CHR$(34):GOTD9803
9852 IFL=32THEN GO SUB 9968:RS!(K)=LEFT$(RS!(L),RS!(SR)):GOTD9803
9853 IFL=33THEN GO SUB 9968:RS!(K)=RIGHT$(RS!(L),RS!(SR)):GOTD9803
9854 IFL=34THEN GO SUB 9966:RS!(N)=MID$(RS!(K),RS!(L),RS!(SR)):GOTD9803
9855 IFL=35THEN GO SUB 9970:RS!(L)=RS!(SR):GOTD9803
9856 IFL=36THEN GO SUB 9970:RS!(L)=RS!(SR):GOTD9803
9857 IFL=37THEN GO SUB 9970:RS!(L)=EXP(RS!(SR)):GOTD9803
9858 IFL=38THEN GO SUB 9970:RS!(L)=LOG(RS!(SR)):GOTD9803
9859 IFL=39THEN S983
9860 IF B7=1 THEN S940
9861 IF B8=1 THEN S940
9862 I7=I7+1:GOTD9803
9863 IFL=40THEN S985
9864 IF GT=1 THEN S940 ELSE S962
9865 IFL=41THEN S987
9866 IFL=41THEN S940 ELSE S962
9867 IFL=42THEN GO SUB 9970:RS!(L)=ASC(RS!(SR)):GOTD9803
9868 IFL=43THEN GO SUB 9970:RS!(L)=CHR$(RS!(SR)):GOTD9803
9869 IFL=44THEN GO SUB 9970:RS!(L)=LEN(RS!(SR)):GOTD9803
9870 IFL=45THEN GO SUB 9970:RS!(L)=SQR(RS!(SR)):GOTD9803
9871 IFL=46THEN I7=R7:RT=0:GOTD9803
9872 IFL=47THEN GO SUB 9970:PRINT RS!(L),RS!(SR):GOTD9803
9873 IFL=48THEN GO SUB 9970:N=RS!(L):K=RS!(SR):SET(N,K):GOTD9803
9874 IFL=49THEN GO SUB 9970:N=RS!(L):K=RS!(SR):RESET(N,K):GOTD9803
9875 IFL=50THEN GO SUB 9970:N=RS!(L):K=RS!(SR):OUTN,K:GOTD9803
9876 IFL=51THEN GO SUB 9970:N=RS!(L):K=RS!(SR):OUTN,K:GOTD9803
9877 IFL=52THEN GO SUB 9972:RS!(SR)=RD(0):GOTD9803
9878 IFL=53THEN S9865
9879 GO SUB 9968:N=RS!(SR):MA$=RS!(K):MB$=RS!(L)
9880 GO SUB 9970:RS!(N)=#0
9881 GOTD9803
9882 IFL=54THEN I1140
9883 IFL=55THEN RT=I7+1:GOTD9840
9884 IFL=56THEN S9866
9885 IFL=57THEN I1140
9886 IFL=58THEN RT=I7+1:GOTD9840
9887 IFL=59THEN S9866
9888 GO SUB 9968:N=RS!(L):SL=RS!(SR):IF POINT(N,SL)THEN RS!(K)=1ELSE RS!(K)=0
9889 GOTD9803
9890 IFL=599THEN I7=I7+1:GOTD9803
9891 PRINT"INVALID OP":N=I7-P:PRINT"LOC":N" CODE":L
9892 GOTD01140
9893 I7=I7+1:N=R7(I7)
9894 I7=I7+1:K=R7(I7)
9895 I7=I7+1:L=R7(I7)
9896 I7=I7+1:SR=R7(I7)
9897 RETURN
9898

```

\*\*\*\*\*  
JOIN THE 1000'S WHO ARE ENJOYING

THE 80 NOTEBOOK

THE MONTHLY JOURNAL FOR ALL TRS-80 USERS

WITH THESE EXCITING FEATURES AND DEPARTMENTS: (AND MORE!!)

|  |                        |
|--|------------------------|
| * SCIENTIFIC SOFTWARE  | * GAMES                |
| * LETTERS TO THE EDITOR  | * PUZZLES              |
| * PRACTICAL APPLICATIONS   | * CONTESTS             |
| * ENTERTAINMENT PROGRAMS   | * GAMBLING             |
| * WORD PROCESSING SYSTEMS  | * NEW PRODUCTS         |
| * ARTIFICIAL INTELLIGENCE  | * TRS-80 CLUB NEWS     |
| * SYSTEM UTILITY SOFTWARE  | * PERSONAL FINANCE     |
| * RADIO SHACK NEWS RELEASES  | * SOFTWARE EXCHANGE    |
| * DATA BASE MANAGEMENT SYSTEMS                                       | * BUSINESS SOFTWARE    |
| * EDUCATIONAL AND CAI PROGRAMMING                                    | * SORTING TECHNIQUES   |
| * SIMULATIONS AND COMPUTER MODELING                                  | * FREE CLASSIFIED ADS  |
| * GRAPHICS AND ANIMATION TECHNIQUES                                  | * PRODUCT EVALUATIONS  |
| * ASSEMBLY LANGUAGE PROGRAMMING LESSONS                              | * ARTICLES ON HARDWARE |
| * LEVEL I, II AND DISK BASIC PROGRAMMING LESSONS                     |                        |
| * OPERATING SYSTEMS, LANGUAGES AND COMPILER DESIGN                   |                        |
| * PROGRAM LISTINGS (UP TO 24 PAGES IN EVERY ISSUE!!)                 |                        |
| * ARTICLES DEALING WITH UNUSUAL AND INTERESTING USES FOR YOUR TRS-80 |                        |

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation, who is not affiliated with THE 80 NOTEBOOK in any way.

---

THE 80 NOTEBOOK

R. D. #3  
Nazareth, PA 18064  
Phone: 215-759-6873

NAME \_\_\_\_\_  
ADDRESS \_\_\_\_\_  
\_\_\_\_\_

CHECK ONE: NEW        RENEWAL         
1 YEAR SUBSCRIPTION: \$14         
2 YEAR SUBSCRIPTION: \$26         
CANADA/MEXICO: ADD \$5/ YEAR

3 YEAR SUBSCRIPTION: \$36         
SAMPLE COPY: \$2        AIR MAIL \$4         
FOREIGN: ADD \$12/ YEAR

AVOID THE 1980 PRICE INCREASES - SUBSCRIBE TODAY!

\*\*\*\*\*

THE 80 NOTEBOOK  
R.D.#3  
Nazareth, PA 18064

BULK RATE  
U.S. POSTAGE  
PAID  
Stockertown, PA 18083  
Permit No. 8

4/80

3/81